

REMARKS

Status of the Claims

- Claims 1, 3, 5-9, 19, 21, and 23-30 are pending in the Application after entry of this amendment.
- Claims 10-18 and 31-33 are withdrawn from consideration
- Claims 1, 3, 5-9, 19, 21, and 23-30 are rejected by the Examiner.
- Claim 3 is amended by Applicant.
- Claims 28-30 are cancelled without prejudice or disclaimer.

Claim Objections

Claim 3 is objected to for improperly depending on cancelled Claim 2. Applicant amends Claim 3 to depend from Claim 1.

Claim Rejections Pursuant to 35 U.S.C. §112

Claims 1, 3, 5-9, 19, 21, and 23-27 stand rejected under 35 U.S.C. §112, first paragraph as failing to comply with the written description requirement. The present Office Action dated 11/30/2006 states that the claimed subject matter in Claims 1, 19, and 25 reciting “shedding XML values from the binary large object” is new because it is not supported by the instant specification. Applicant respectfully disagrees.

Applicant notes that paragraph 0005 of the as-filed specification includes the statement:

“The XML index is implemented as a node table wherein the population of the node table may be performed by shredding the input XML values stored as binary large objects in an XML column of the primary table.” (from paragraph 0005).

Thus, Applicant notes that one aspect of the invention includes shredding XML values that are stored as Binary Large Objects (BLOBS). In another aspect of the invention, the XML values are obtained from the BLOB by shredding the BLOB itself. As indicated in paragraph 0059, commands which include:

“SELECT * ”

```
FROM          Rowset-generated-by-shredding-XML-  
BLOB-@xmlcol-with-  
path-expression-and-max-level-applied  
  
INSERT      INTO @ShreddedRows"  
  
(from paragraph 0059)
```

Thus, Applicant submits that the BLOB itself is shredded and XML values are obtained from the BLOB by the shredding process. This is advantageous because the results of the shredding (parsing) are saved in the populated tables to reduce the search time for subsequent searches as referred to in paragraph 0013 of the present specification. As stated in paragraph 0013:

“In accordance with aspects of the invention, the results of parsing the binary large objects may be saved and future queries may use the results and run significantly faster.” (paragraph 0013)

Accordingly, Applicant respectfully submits that there is support in the specification at paragraphs 0005, 0013, and 0059 for the phrase “shredding XML values from the binary large object” as recited in current Claims 1, 19, and 25. Applicant respectfully requests withdrawal of the 35 U.S.C §112 rejection of Claims 1, 3, 5-9, 19, 21, and 23-27.

Claim Rejections Pursuant to 35 U.S.C. §102

Claims 28-30 stand rejected pursuant to 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,584,459 to Chang et al. (Chang).

Applicant cancels Claims 28-30 without prejudice or disclaimer.

Claim Rejections Pursuant to 35 U.S.C. §103 (a)

Claims 1, 3, 5-9, 19, 21, 23-24, and 25-27 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,584,459 to Chang et al. (Chang) in view of U.S. Patent No. 7,096,224 to Murthy et al. (Murthy). Applicants respectfully traverse the rejection.

Chang teaches:

“In accordance with the present invention, an XML extender for a computer-implemented relational database system is disclosed for storing, querying, and retrieving structured documents. Generally, relational extenders define and implement complex data types and extend the tables within the relational database with the new data types. The XML extender provides a new Abstract Data Type (ADT) DB2XML, which can be specified as a column data type, and includes several User Defined Functions (UDFs) for storing, searching, and retrieving XML documents internally, as DB2.RTM. Character Based Large Objects (CLOB), or externally, in flat files or Uniform Resource Locators (URLs), for example.”
(col. 3 lines 48-60).

Applicant submits that Chang teaches an extender that provides a new abstract data type for storing querying, and retrieving structured documents internally as CLOBs or externally, in flat files or URLs. (See Chang, Abstract). Applicant notes that a CLOB and a BLOB are different data types and that Chang teaches the use of a CLOB with an extender that defines a new data type. Whereas Claim 1 recites that a primary table structure holds XML data as a BLOB and that the BLOB is shredded to populate the XML index, Chang teaches no such functionality.

Claim 1, in relevant part, recites:

A computer-executed method of retrieving XML data from a database, the method comprising:

creating a primary table structure to hold XML data as a binary large object...;

creating a primary XML index relating to the primary table structure, where the primary XML index includes a node table;

populating the primary table and the primary XML index, wherein the primary XML index is populated by shredding XML values from the binary large object and storing the values into the primary XML index, and wherein the XML index preserves document order and structure;

querying on the primary table...; and

retaining the primary table and primary XML index so that subsequent queries execute faster than an initial query.

Applicant notes that Chang fails to teach:

- (1) populating an XML index by shredding XML values from a large binary object (BLOB) and storing the values into the primary XML index which preserves document order and structure;
- (2) retaining the primary index, having the shredded XML values from the BLOB so that subsequent queries execute faster than an initial query.

Applicant suggests that the addition of Murthy to Chang does not cure the deficiency of teaching of Chang because Murthy also fails to teach element (1) and element (2) referenced above.

Murthy teaches:

“According to one aspect, a mechanism is provided to allow users of a database system to register XML schemas with the database system. An XML schema may be registered explicitly (via an API call) or implicitly (when an instance document conforming to the XML schema is first inserted into the database).

During the registration process for a given XML schema, the database system determines (1) an appropriate database representation for the XML schema and (2) mapping information. The "appropriate database representation" determination is a determination about how data that conforms to the XML schema should be managed by the database system.” (col. 3, lines 33-44).

Applicant submits that Murthy, like Chang, does not teach elements (1) and (2) as mentioned above. Further, the combination of Chang and Murthy do not teach the above-mentioned elements of (1) populating an XML index by shredding XML values from a large binary object (BLOB) and storing the values into the primary XML index which preserves document order and structure; and (2) retaining the primary index, having the shredded XML values from the BLOB so that subsequent queries execute faster than an initial query.

The combination of Chang and Murthy cannot render obvious independent Claims 1, 19, and 25 under 35 U.S.C. § 103(a) because all of the elements of the pending independent claims are not taught by the combination. This conclusion comports with MPEP § 2143.03

DOCKET NO.: MSFT-2160/304750.01
Application No.: 10/693,750
Office Action Dated: November 30, 2006

**PATENT
REPLY FILED UNDER EXPEDITED
PROCEDURE PURSUANT TO
37 CFR § 1.116**

where all elements of the pending claims must be taught in the combination of the references in order to construct a prima facie case of obviousness.

Conclusion

In light of the amendments and discussion above, Applicant respectfully submits that all pending claims patentably define over the cited art. Applicant respectfully requests reconsideration and withdrawal of the rejections of all pending claims.

Respectfully submitted,

Date: January 30, 2007

/Jerome G. Schaefer/

Jerome G. Schaefer
Registration No. 50,800

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439